Trusted Computing

Lukas Rupprecht
Betreuer: Holger Kinkelin
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: rupprech@in.tum.de

***Kurzfassung*— 1999 wurde die Trusted Computing Platform Alliance (heute: Trusted Computing Group) gegründet, ein Zusammenschluss großer Unternehmen, deren Ziel es ist, den Ansatz des Trusted Computing voranzutreiben und im Anwenderbereich zu etablieren. Trusted Computing ist ein Sicherheitsmechanismus, der durch eine im System verankerte Vertrauenswurzel, dem Trusted Platform Module, erreichen soll, dass das System eindeutig identifizierbar und attestierbar wird. Das bedeutet, dass es keine Möglichkeiten mehr geben soll, die Identität einer Plattform zu fälschen und deren aktuellen Systemzustand (laufende und installierte Software, Konfigurationen) vertrauenswürdig festzustellen. Durch die in den Spezifikationen der Trusted Computing Group vorgeschlagenen Mechanismen zur Umsetzung dieser Eigenschaften eröffnen sich jedoch für die Hersteller solcher Module und der zugehörigen Software eine Reihe an Kontrollmöglichkeiten über die Plattformen, was zu großer Kritik an dem Gesamtkonzept geführt hat. Diese Arbeit ist ein Einstiegspunkt in die Thematik und erklärt die grundlegenden Mechanismen und Eigenschaften einer Trusted Platform sowie den Aufbau und die Funktionsweise der notwendigen Komponenten. Auch die Pro-Kontra-Frage wird beidseitig beleuchtet.**
***Schlüsselworte*— Trusted Computing, Trusted Computing Group, Trusted Platform Module, Trusted Software Stack**

I. Einleitung

In der heutigen Zeit, in der fast jeder Rechner, vom Heim PC bis zum Handy, in irgend einer Art und Weise mit dem Internet verbunden ist, wurde Computersicherheit zu einem Thema, welches für jeden Anwender, sei es Großunternehmer oder Privatmann, eine große Rolle spielt. Immer neue Techniken und Lücken werden entdeckt, die es ermöglichen, in ein System einzudringen und dort Schaden zu verursachen. Viren, Würmer und Hacker kennt fast jeder, aber auch Begriffe wie Social Engineering gewinnen immer mehr an Bedeutung. Zum Schutz vor all diesen Gefahren wurden Systeme entwickelt, welche von einem kleinen Freeware Virenscanner bis hin zum komplexen Firewallsystem aus Soft- und Hardwarekomponenten reichen und versuchen, die eigenen Daten zu sichern. Ein Ansatz, um dieses Ziel zu erreichen, ist das *Trusted Computing*. Nach der *Trusted Computing Group* (TCG) bedeutet Trust hier: „Trust is the expectation that a device will behave in a particular manner for a specific purpose." ([12], S. 15), also die Erwartung oder das Vertrauen, dass die Plattform für einen bestimmten Zweck ein bestimmtes Verhalten aufweist. Das ist allerdings nicht die einzige Definition. Seit 20 Jahren wird dieser Begriff nun schon verwendet und je nach Auslegung anders definiert. So definiert die NSA Trust beispielsweise als: „A Trusted System or component is one whose failure can break the security" ([12], S. 14), also eher im Sinne von „für die Sicherheit verwantwortlich". Das Ziel des Trusted Computing (nach TCG) ist es, Rechensysteme so vertrauenswürdig zu machen, dass es unmöglich wird, deren Identität zu fälschen um sich somit gegenüber Kommunikationspartnern eindeutig zu identifizieren. Ein Mechanismus zum sicheren, lokalen Speichern von sensiblen Daten und die Möglichkeit, Plattforminformationen über Softwarekomponenten zu erhalten sind ebenfalls Teil dieser Initiative. So ein Eingriff in die Architektur kann natürlich auch Einschränkungen für den Benutzer mit sich bringen und bietet die Möglichkeit des Missbrauchs, da die Kontrollmöglichkeiten über die Plattform steigen. Dies führte zu einer beträchtlichen Bewegung gegen das Trusted Computing.

Diese Arbeit erläutert die Grundzüge des Trusted Computing, welche durch die Trusted Computing Group spezifiziert werden und geht dabei vor allem auf die dort verwendeten Techniken ein. Kapitel II stellt die Trusted Computing Group und deren Organisation vor. In Kapitel III werden die zum Verständnis notwendigen Grundlagen bereitgestellt. Kapitel IV beschäftigt sich mit dem Trusted Platform Module und Kapitel V führt den Trusted Software Stack ein. Kapitel VI beschreibt die Remote Attestation als Beispielanwendunge einer Trusted Platform und Kapitel VII beleuchtet die Argumente der Gegner und der Befürworter. In Kapitel VIII wird ein kleiner Ausblick auf zukünftige Entwicklungen sowie ein Fazit gegeben.

II. Die Trusted Computing Group

Trusted Computing ist ein Begriff, der schon seit längerer Zeit existiert und, wie so viele andere Innovationen im Kommunikations- und Securitybereich, durch das Militär entstanden ist. Hört man heutzutage diesen Begriff verbindet man mit ihm allerdings eher die Trusted Computing Group[1] und deren Arbeit. Die Trusted Computing Group ist ein Zusammenschluss von Unternehmen, welche den Ansatz des Trusted Computing im zivilen Bereich und für den Privatanwender weiterentwickeln und standardisieren will. Sie entstand 2003 aus der 1999 gegründeten *Trusted Computing Platform Alliance*, die ein Zusammenschluss der Firmen Microsoft, IBM,

[1] http://www.trustedcomputinggroup.org

Hewlett Packard und Compaq war. [6] Heute gehören ihr ca. 170 Unternehmen weltweit an, darunter z.B. Intel, AMD, Infineon, Motorola oder Nokia. Die TCG bezeichnet sich selbst als eine „not-for-profit organization" mit dem Ziel, Standards zu entwickeln und zu veröffentlichen, um Trusted Computing zu einem wesentlichen Bestandteil moderner Rechensysteme zu machen. Sie stellt Spezifikationen für die Hauptbestandteile eines *Trusted Computing System* (TCS) bereit, erarbeitet grundlegende Konzepte, welche frei einsehbar und somit auch frei umsetzbar sind und fördert dadurch deren Einsatz und deren Verwendung. Der TCG angehörige Unternehmen haben es sich zum Ziel gemacht, ihre entworfenen Standards auch konkret zu implementieren und so gab und gibt es schon einige Systeme, die die geforderten Funktionalitäten unterstützen. Das von Microsoft entwickelte Palladium bzw. NGSCB ist ein Beispiel für die Umsetzung eines *Trusted Operating System* und Firmen wie Lenovo oder auch Infineon liefern bereits die für Trusted Platfformen notwendige Hardware (das sog. *Trusted Platform Module* (TPM)). Trusted Computing ist jedoch nicht allein die TCG und deren Standards sind keine Dogmen, die bei der Entwicklung eines Trusted Computing System erfüllt werden müssen. Allerdings sind deren Spezifikationen vorreitend und maßgebend für die Umsetzung solcher Systeme und deswegen wird in dieser Arbeit auch auf Trusted Computing in der Form der TCG eingegangen. Im Folgenden wird betrachtet, was konkret ein Trusted Computing System ist und welche Eigenschaften und Anforderungen es vorweisen und erfüllen muss.

III. Grundlagen und Konzepte

Ein TCS basiert auf vielen grundlegenden Techniken und Ansätzen aus der Kryptologie, um ein System vertrauenswürdig machen zu können. Die Grundlagen, die in einem TCS umgesetzt werden und die zum Verständnis notwendig sind, werden nun kurz erläutert.

A. Public Key Infrastrukturen

Ein Konzept, welches in einem TCS eingesetzt wird, sind Public Key Infrastrukturen oder PKI's. Diese werden heutzutage häufig eingesetzt, wenn es um verschlüsselte, sichere Nachrichtenübertragung geht. Eine PKI setzt ein Public Key Kryptosystem voraus. Ein solches System verwendet einen privaten Schlüssel zum Entschlüsseln von Nachrichten und einen zugehörigen öffentlichen Schlüssel zum Verschlüsseln derselben. RSA beispielsweise ist ein populäres Verfahren dieser Gattung. Kurz zusammengefasst funktionieren PKI's folgendermaßen:
Es existiert ein privater Schlüssel (Private Key), welcher mit Hilfe eines Algorithmus (z.B. RSA) erzeugt wurde. Aus diesem lässt sich nun ein öffentlicher Schlüssel berechnen. Dieser Public Key ist frei verfügbar und jeder, welcher mit dem Besitzer des privaten Schlüssels sicher kommunizieren möchte, kann den Public Key verwenden um Nachrichten zu verschlüsseln.
Das Wichtigste an solche Verfahren ist, dass es bei ausreichenden Schlüssellängen nicht möglich ist, aus dem öffentlichen Schlüssel den privaten Schlüssel zu ermitteln und dass chiffrierte Nachrichten nur mit dem privaten Schlüssel entschlüsselt werden können.

B. Vertrauenskette und Vertrauenswurzel

Ein weiterer Ansatz ist die Vertrauenskette (Chain of Trust) mit der Vertrauenswurzel (Root of Trust) als Ausgangspunkt. Die Idee dahinter lässt sich am besten anhand der oben beschriebenen PKI's erläutern: Ein Problem, welches sich bei PKI's ergibt, ist die Bereitstellung des öffentlichen Schlüssels. Für einen Nutzer dieses Schlüssels muss gewährleistet werden, dass der Schlüssel, den er verwenden will, wirklich auch der ist, welcher von dem gewünschten Kommunikationspartner bereitgestellt wurde. Um dies sicherzustellen und die Integrität und Gültigkeit der Informationen zu validieren, werden *Zertifikate* eingesetzt. [3] Ein solches Zertifikat verpackt die Informationen über den Besitzer und dessen öffentlichen Schlüssel und stellt diese, wiederum verschlüsselt, zur Verfügung. Um nun an die enthaltenen Daten zu gelangen, wird wieder ein Schlüssel benötigt. Dieser wird von *Certification Authorities* (CA's), welche die Zertifikate erstellen, bereitgestellt. Zwischen Sender und Empfänger können nun beliebig viele Zertifikate existieren, die die Authentizität des darunter liegenden sicherstellen. Der Sender, der ja den Public Key benötigt, muss sich durch diese Hierarchie hangeln, bis eine CA erreicht ist, der er vertraut. So entsteht also eine Kette aus Zertifikaten. Das oberste Glied in dieser Kette nennt man Root CA. Hieran erkennt man nun sehr gut, nicht nur wegen der begrifflichen Parallelen, dass Konzept der Vertrauenskette. Ein Zertifikat wird verwendet, um die Vertrauenswürdigkeit eines darunter liegenden Zertifikates zu gewährleisten. Die Root CA, welche die Vertrauenswurzel bildet, hat keine darüber liegende Instanz mehr, welche deren Vertrauenswürdigkeit bestätigt, und somit muss man dieser von sich aus vertrauen. Beim Trusted Computing geht es nun darum, nicht die Vertrauenswürdigkeit eines Empfängers, sondern die einer Plattform zu gewährleisten. Hierzu muss also eine Vertrauenswurzel im System verankert werden, von welcher ausgehend sich das restliche System überprüfen lässt.

C. Plattform Attestation und Authentication

Der dritte Punkt beschäftigt sich mit der Bewertung (Attestation) und Identifizierung (Authentication) einer Plattform. [14] Genau genommen sind dies zwei verschiedene Kriterien, da sie jedoch ähnlich umgesetzt werden, sind sie hier in einem Unterpunkt zusammengefasst.
Unter *Attestation* versteht man die Bewertung eines Systems. Bewertet wird die Vertrauenswürdigkeit nach Kriterien wie ausgeführter und installierter Software sowie verschiedenen Konfigurationsdateien. Hierfür muss der Zustand eines Systems festgehalten und protokolliert werden. Diesen Vorgang bezeichnet man als *Integritätsmessung* (Integrity Measurement). Das bedeutet, dass über ausgewählte Systemkomponenten und Programme ein SHA-1 Hashwert berechnet und gespeichert wird. Um eine verlässliche Messung zu liefern, muss diese von der Vertrauenswurzel des Systems ausgehen,

da nur so sichergestellt werden kann, dass weitere Messungen nicht verfälscht wurden.
Authentication bezeichnet den Vorgang der Identitätsbestimmung. Die Plattform muss sich gegenüber einem Dritten authentifizieren um zu beweisen, dass sie die ist, für die sie gehalten wird. Auch dieser Vorgang erfordert eine Vertrauenswurzel, von der ausgehende die Identifizierung stattfinden kann.
Die Vertrauenswurzel soll es unmöglich machen, dass falsche Informationen in der Vertrauenskette weitergereicht werden und so Möglichkeiten bieten könnten, Identitäten zu fälschen oder Systemzustände vorzutäuschen, die so gar nicht vorhanden sind. Ein Ziel des Trusted Computing besteht also darin, ein System eindeutig identifizierbar zu machen. Es soll nicht möglich sein, dass das System sich als etwas ausgibt, was es nicht ist. Im folgenden werden die einzelnen Komponenten einer Trusted Plattform erläutert

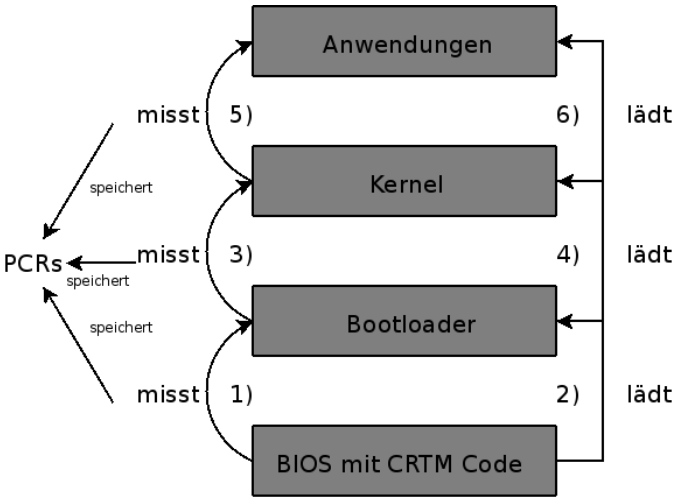

Abbildung 1. Vertrauenskette eines Trusted Bootvorgang

IV. Das Trusted Platform Module

Um die oben erwähnten Konzepte für eine Plattform umzusetzen wird spezielle Hardware verwendet, das *Trusted Platform Module* (TPM). [2] [13] Es ähnelt einer SmartCard, nur dass es nicht an einen Benutzer sondern an ein System gebunden ist. Das TPM ist ein Mikrocontroller und bildet die Vertrauenswurzel des Systems. In ihm werden alle notwendigen Funktionen bereitgestellt, die die Plattform sicher und vertrauenswürdig machen sollen. Im Folgenden werden die einzelnen Bestandteile eines TPM vorgestellt.

A. Die Cryptoengine

Ein Trusted Platform Modul besitzt integrierte Funktionen, um kryptographische Berechnungen auszuführen. Dazu zählen unter anderem ein RSA Schlüssel Generator, ein „echter" Zufallszahlengenerator oder integrierte Berechnungen von Hashfunktionen wie z.B. SHA-1. Durch die Kapselung dieser Funktionen im TPM wird erreicht, dass keine sensiblen Informationen das Modul verlassen müssen.

B. Die Core Root of Trust for Measurement

Die *Core Root of Trust for Measurement* (CRTM) wird zur Integritätsmessung in einem System verwendet. Die daraus resultierenden Messwerte (die Hashwerte) werden im TPM (in den sog. *Platform Configuration Registers* (PCR's)) abgelegt. Die CRTM befindet sich im BIOS und wird als erste Komponente beim Bootvorgang geladen. Die Grundidee ist hierbei, dass ausführbarer Code und Konfigurationsdateien gemessen werden bevor sie geladen werden. Dadurch wird ein Trusted Boot Vorgang realisiert (s. Abb. 1). Dieser beginnt mit dem Laden des CRTM und der Messung der ersten Komponente (normalerweise dem BIOS). Von diesem Punkt aufsteigend wird eine Vertrauenskette gebildet, bei der die einzelnen Softwarekomponenten von ihren darunter liegenden Komponenten gemessen und dann ausgeführt werden. Ein Trusted Bootloader (z.B. Trusted Grub[2]) beginnt damit, den Code, der zum Starten des Betriebssystems notwendig ist, Stück für Stück zu laden. Vor dem Laden eines Teils wird dieser vermessen, danach geladen und ausgeführt, dann der nächste Teil gemessen usw. bis der komplette Kernel geladen und das Betriebssystem ausführbar ist. Die Werte werden im TPM protokolliert. Da die Messungen bei der CRTM starten und diese sich im TPM befindet, also eine vertrauenswürdige Wurzel darstellt, kann eine lückenlose Vertrauenskette aufgebaut werden, in der keine Manipulationen oder Fälschungen möglich sein sollen.

[2]http://trousers.sourceforge.net/grub.html

C. Der Protected Storage

Das TPM stellt einen geschützten Speicherbereich zur Verfügung, in dem geheime Daten wie Schlüssel, Passwörter und auch die Hashwerte der CRTM abgelegt werden können. Um die Daten verschlüsselt speichern zu können wird eine Schlüssel-Hierarchie benötigt, welche den *Storage Root Key* als Wurzel hat. Diese Konzept wird in Abschnitt IV-D.3 genauer betrachtet.

D. Schlüssel und Zertifikate

Neben der Hard- und Firmware eines TPM existieren einige essentielle Daten, die die Eindeutigkeit des Moduls und somit der Plattform garantieren, und die zur Umsetzung der Anforderungen benötigt werden. Die folgenden Auflistung stellt diese vor. [4]
1) Der Endorsement Key: Der *Endorsement Key* (EK) ist die Vertrauenswurzel des Trusted Platform Moduls. Er ist ein RSA-Schlüssel mit einem privaten und einem öffentlichen Teil und immer nur genau einer Plattform zugeordnet. Er wird bei der Aktivierung des TPM erzeugt und der private Teil verlässt dieses auch nie. Er kann auch nicht auf ein anderes System übertragen werden. Mit Hilfe des EK wird die spezifikationsgerechte Funktionsweise des zugehörigen TPM garantiert. Neuere Spezifikationen (v. 1.2) erlauben zwar die Erstellung anderer EK's, dies führt allerdings zum Vertrauensverlust, da die durch den EK garantierten Eigenschaften nun nicht mehr gegeben sein müssen.

2) Attestation Identity Keys: Attestation Identity Keys (AIK's) sind Schlüsselpaare, die nach Bedarf im TPM erzeugt werden und zur Plattformauthentifizierung und -attestierung benutzt werden. Sie sind notwendig, damit Daten nicht mit dem Endorsement Key signiert werden müssen. Dieses Konzept löst ansatzweise ein durch die Eindeutigkeit des EK bedingtes Privacy Problem. Würden nämlich Dokumente mit diesem signiert, wäre die Signatur und somit das Dokument genau einer Plattform zuzuordnen und die Anonymität des Benutzers ginge damit verloren. Eine Beschreibung der genauen Erzeugung und Funktion der AIK's wird in Abschnitt VI gegeben.

3) Storage Root Key: Der *Storage Root Key* (SRK) wird ähnlich wie der Endorsement Key im TPM angelegt und verlässt dieses niemals. Er ist zuständig für die Verwaltung und den Zugriff auf den geschützten Speicherbereich (s. IV-C) des TPM und stellt Schlüssel zur Ver- und Entschlüsselung dort abgelegter Daten bereit. Der private Teil des SRK ist das oberste Element der TPM Key Hierarchie. Wenn ein neuer Schlüssel zum geschützten Ablegen von Daten benötigt wird, kann dieser im TPM erzeugt werden. Um den Schlüssel selbst zu sichern wird dieser nun mit dem in der Hierarchie über ihm liegenden Schlüssel verschlüsselt. Somit ist sicher gestellt, dass die Daten nur mit Hilfe des TPM wieder entschlüsselt werden können (Eine solche Hierarchie befindet sich in Abb. 2). So verschlüsselte Daten werden entsprechend *Key-Blobs* (Blob = binary large object) oder *Data-Blobs* genannt. Allgemein bezeichnet man diese als *TPM protected objects*.
Eine Besonderheit des TPM ist auch, dass Daten logisch an die Plattform gebunden werden können (*Sealing*). Hierbei kann bei der Erstellung eines TPM protected objects der aktuelle Systemzustand, welcher bei der CRTM Messung festgehalten wurde, mit in die Verschlüsselung einbezogen werden. Folglich können solche Objekte nur wieder entschlüsselt werden, wenn der Systemzustand exakt dem Zustand während der Erstellung entspricht. Diesen Mechanismus bezeichnet man als *Sealed Storage*.

4) Zertifikate: Ein TPM muss zusätzlich zu den oben genannten Schlüsseln noch die drei, in Zertifikatform vorliegenden, Informationen enthalten:

- Endorsement Credential
- Platform Credential
- Conformance Credential

Im Endorsement Credential wird der öffentliche Teil des Endorsement Key bereitgestellt. Es bestätigt die Authentizität der Plattform. Das Platform Credential stellt sicher, dass die erforderlichen Plattformkomponenten (nach Spezifikation) vorhanden und validiert sind und im Conformance Credential wird garantiert, dass das System wie erwartet funktioniert.
Nachdem nun die Hardwarekomponenten eines Trusted Computing System abgehandelt sind, geht es im nächsten Teil darum, wie ein Betriebssystem bzw. Software die bereitgestellten Funktionen nutzen kann.

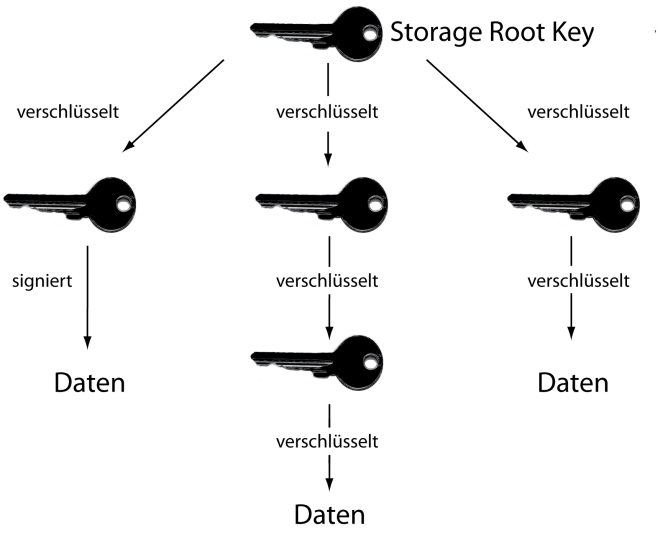

Abbildung 2. TPM Key Hierarchie

V. Softwareunterstützung und Trusted Software Stack

Die bis jetzt vorgestellten Funktionen eines TPM waren alle passiv. Es wurde nur gemessen, protokolliert und Möglichkeiten wie geschützter Speicher zur Verfügung gestellt. Um ein TPM auch aktiv nutzen zu können, benötigt man neben Hard- und Firmware auch noch eine Softwarekomponente. Diese bezeichnet man als den *Trusted Software Stack* (TSS). [20] Über ihn können das Betriebssystem und laufende Anwendungen mit dem TPM kommunizieren und dessen Dienste in Anspruch nehmen. Der TSS besteht aus mehreren Komponenten:

- Die unterste Komponente ist der TPM Treiber. Anwendungen können nur über diesen mit dem TPM kommunizieren und es darf keine Möglichkeit geben, ihn zu umgehen. Er bildet somit die einzige Schnittstelle zum TPM.
- Auf den Treiber aufsetzend folgt die TDDL (TCG Device Driver Library). Diese stellt eine homogene Schnittstelle für alle TPM's zur Verfügung und bildet den Übergang zwischen User und Kernel Mode.
- Die TSS Core Services (TCS) machen Anwendungen alle Grundfunktionen des TPM zugänglich. Sie kommunizieren mit dem TPM über das TDDLI (TCG Device Driver Library Interface)
- Die TSS Service Providers (TSP) sind für die Nutzung der kompletten Möglichkeiten eines TPM zuständig. Die Kommunikation mit dem TPM erfolgt über das Trusted Software Stack Core Services Interface. Die Service Providers stellen für die eigentlichen Anwendungen das TSPI (TSS Service Providers Interface) bereit, über das diese dann auf das TPM zugreifen können.

In Abb. 3 ist der Aufbau noch einmal grafisch dargestellt. Es existieren bis jetzt schon einige Implementierungen eines TSS. Eine kommerzielle Implementierung wird z.B. von

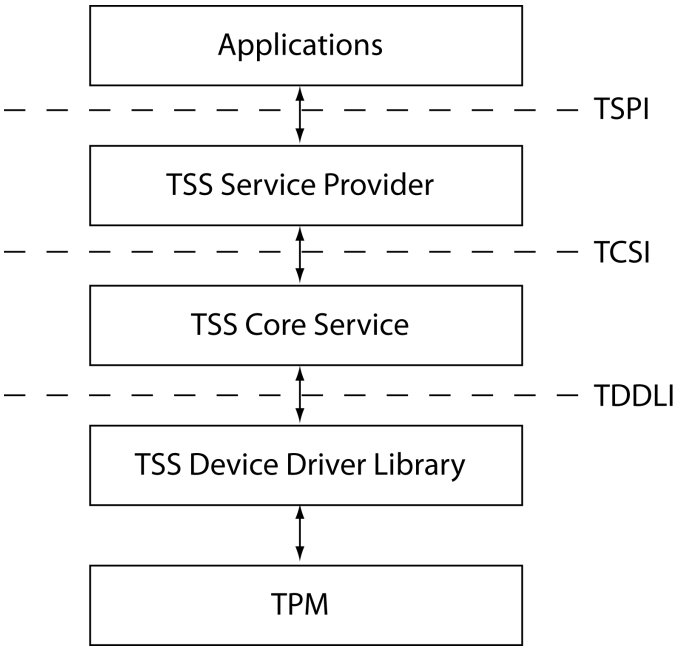

Abbildung 3. Aufbau des TSS

STMicroelectronics angeboten[3]. TrouSerS[4] oder das an der TU Graz entwickelte Trusted Java[5] sind Open Source Varianten eines TSS. Mit Hilfe der Softwareunterstützung kann nun ein gängiges Anwendungsbeispiel von Trusted Computing Systemen betrachtet werden.

VI. Remote Attestation als Beispielanwendung einer Trusted Plattform

In vielen Fällen ist es wichtig zu wissen, wie genau der Kommunikationspartner aussieht. Man möchte z.B. wissen, ob das gegenüberliegende System aktuelle Software mit aktuellen Patches enthält, oder ob noch veraltete, unsichere Versionen verwendet werden. Oder man möchte sichergehen, dass keine kompromittierte Software auf dem System vorhanden ist. Und natürlich möchte man auch sicher sein, dass der Kommunikationspartner keine gefälschte Identität verwendet und jemand ganz anderes ist. Mit herkömmlichen Mitteln ist dies nur bedingt möglich, da man zwar Zertifikate zur Clientauthentifizierung verwenden kann, allerdings über den Zustand des Clientsystems wenig bis gar keine Informationen besitzt. Trusted Computing bietet hierfür einen Lösungsansatz, die *Remote Attestation*. Wie bereits in Abschnitt III-C erläutert, wird hier ein Mechanismus bereitgestellt, der es ermöglicht, den genauen Zustand eines Kommunikationspartners festzustellen und zwar mit der Sicherheit, keine gefälschten Informationen zu erhalten. Die genaue Funktionsweise soll in diesem Abschnitt am Beispiel der von IBM implementierten *IMA*[6] (Integrity Measurement Architecture) unter Linux erklärt werden [15], welche die TPM-internen Messmechanismen auf das laufende Betriebssystem erweitert und somit das System dynamisch vermessen und protokollieren kann. Remote Attestation ist auch ohne Softwareunterstützung möglich, allerdings reichen die bereitgestellten Funktionen des TPM dann nur bis zur Vermessung des Kernels. Im Folgenden wird der Kommunikationspartner, der Informationen eines Systems anfordert als *Verifier* (Prüfer) bezeichnet, das System, welches Informationen über sich liefern soll, wird *Attesting Party* (zu beglaubigender Teilnehmer) genannt.

A. Das Attestation Identity Key Konzept

Das erste Problem welches Auftritt, wenn sensible Daten verschickt werden, ist deren Integrität und deren Authentizität zu gewährleisten. Diese Eigenschaften werden normalerweise mit digitalen Signaturen und Zertifikaten bestätigt. Auch bei der Remote Attestation werden verschickte Messwerte so validiert. Allerdings, wie bereits in IV-D.2 erwähnt, kommt es zu Privacy Problemen, wenn die Signaturen mit dem Endorsement Key erstellt werden und als Authentifikation das Endorsement Credential bereitgestellt wird, da der EK eindeutig ist und somit die Anonymität verletzt werden würde. Um diese Probleme zu lösen wurden die Attestation Identity Keys (AIKs) eingeführt. [2] [4] Mit ihnen können vertrauenswürdige Signaturen erstellt werden und jeder AIK erhält bei seiner Erstellung ein Zertifikat um seine Authentizität zu garantieren. Das Anlegen eines neuen AIK ist in Abb. 4 dargestellt. Hierbei erstellt das TPM zuerst ein neues Schlüsselpaar (den AIK) und schickt den öffentlichen Teil zusammen mit dem Endorsement Credential, dem Platform Credential und dem Conformance Credential an einen vertrauenswürdige Privacy CA. Die Daten werden mit dem privaten Teil signiert, um die Verbindung von privatem und öffentlichem Schlüssel zu gewährleisten. Die Privacy CA validiert nun die erhaltenen Daten und die Signatur. Ist die Prüfung erfolgreich, also sind Endorsement Credential, usw. gültige Zertifikate, erstellt sie für den neu angelegten AIK ein *Identity Credential* und schickt dieses, verschlüsselt mit dem öffentlichen Teil des Endorsement Key, zurück an das TPM. Somit ist der AIK jetzt vertrauenswürdig und kann zum signieren von Daten verwendet werden. Da ein AIK jedoch nicht mehr eindeutig ist, ist er auch nicht mehr genau einer Plattform zuordenbar. Kritiker bemängeln jedoch, dass dadurch nur eine weitere Instanz zwischengeschaltet wurde, dass eigentliche Problem aber bestehen bleibt, da die Privacy CA immer noch eine eindeutige Zuordnung durchführen kann. [1]

B. Die Integrity Measurement Architecture für Linux

Nachdem es jetzt möglich ist, Informationen zwischen Verifier und Attesting Party auszutauschen, ohne die Anonymität der Attesting Party zu verletzen, kann nun die eigentliche Remote Attestation stattfinden. Deren Ziel ist es, die laufende Software der Attesting Party vertrauenswürdig festzustellen, um so eine Aussage über die Sicherheit und die Vertrauenswürdigkeit von dieser machen zu können. Hierzu werden ein TPM und die entsprechende Softwareunterstützung, die IMA, benötigt,

[3] http://www.st.com/stonline/products/literature/bd/10928.htm
[4] http://trousers.sourceforge.net/
[5] http://trustedjava.sourceforge.net/
[6] http://domino.research.ibm.com/comm/research_projects.nsf/pages/ssd_ima.index.html

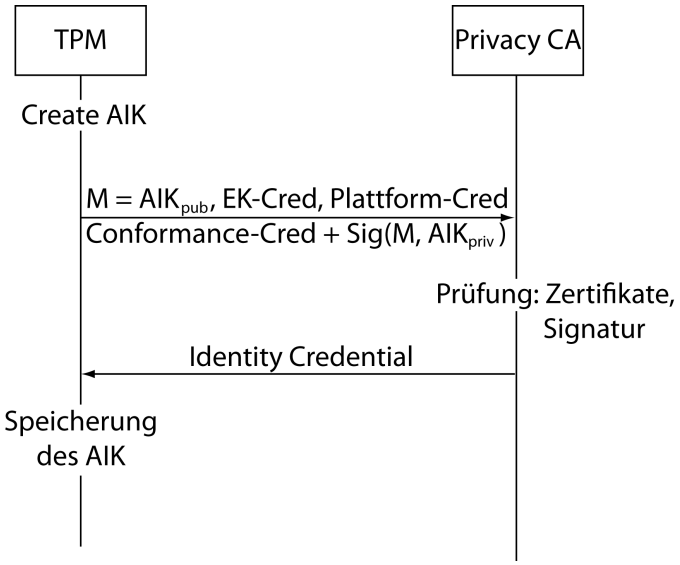

Abbildung 4. Erstellung eines neuen AIK

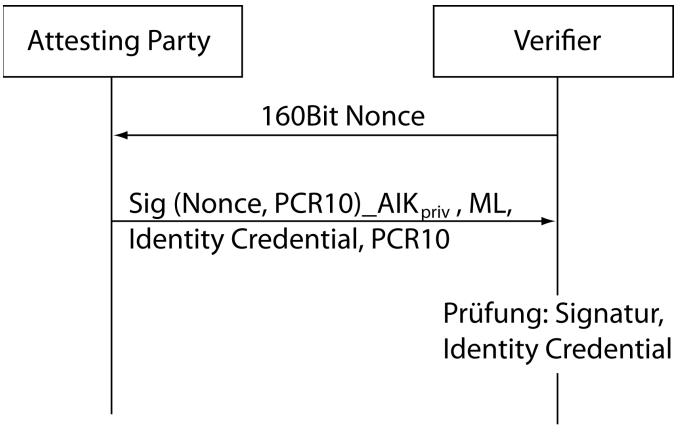

Abbildung 6. Das Integrity Challenge Protocol

um auch die laufenden Anwendungen mit in die Messungen einzubeziehen. [15]
Die IMA besteht aus drei Komponenten:

- Measurement Mechanism (MM)
- Integrity Challenge Mechanism (ICM)
- Integrity Validation Mechanism (IVM)

Die drei Bestandteile und deren Funktionen werden im Folgenden erklärt.
1) Der Measurement Mechanism: Dieser ist zuständig für die Messungen im System. Die Idee ist hierbei, dass Bootvorgang und Kernel, wie in Abschnitt IV-B beschrieben, vermessen werden und der MM die Messungen im Laufenden System übernimmt. Dadurch, dass die IMA teilweise im Kernel vorhanden ist und mit dem Betriebssystem geladen wird, ist diese auch vertrauenswürdig. Der MM bildet nun immer, bevor ein Programm ausgeführt werden soll, einen SHA-1 Wert über dieses (Fingerprint) und speichert das Ergebnis in einer *Measurement Liste* (ML). Es können auch bestimmte sensible Konfigurationsdaten und anderer ausführbarer Code gemessen werden, was eine vollständige Protokollierung ermöglicht. Zusätzlich wird bei jedem Schreibvorgang in die Measurement Liste, das PCR10 *extended*. Das bedeutet, dass der eben gemessene Wert an das Register angehängt wird , davon ein SHA-1 berechnet wird und dieses Ergebnis als neuer Wert in PCR10 geschrieben wird. Dadurch werden die Messwerte vor Fälschung geschützt.

```
var newPCR10 = PCR10.concat(measureValue);
newPCR10 = SHA-1(newPCR10);
PCR10 = newPCR10;
```

Abbildung 5. Die Funktion „extend"

2) Der Integrity Challenge Mechanism: Die oben erzeugten Informationen können nun bei Bedarf durch den Verifier abgefragt werden, mit der Sicherheit, dass diese auch wirklich vertrauensvoll sind. Hierzu existiert der ICM, welcher das Integrity Challenge Protocol implementiert. Dies wird benötigt um die Daten auch sicher auszutauschen und sie z.B. gegen Replay Attacken oder Fälschung während der Übertragung zu immunisieren. Die Funktionsweise ist in Abb. 6 dargestellt. Zuerst wird eine 160 Bit Nonce an die Attesting Party geschickt (normalerweise eine Zufallszahl, da diese nicht vorhersagbar sein darf). Die Attesting Party signiert nun mit einem AIK PCR10 und Nonce und schickt diese Signatur zusammen mit Measurement Liste, zugehörigem Identity Credential und PCR10 zurück an den Verifier. Dieser Vorgang wird als *quote* bezeichnet. Über das zugehörige Identity Credential wird der AIK validiert und anschließend die Signatur geprüft. Zusätzlich wird aus der Measurement Liste der PCR10 Wert wie in Abb. 5 berechnet und mit dem erhaltenen PCR10 Wert verglichen. Stimmen diese und die Werte der Nonce überein, war die Prüfung erfolgreich und die Measurement Listen sind vertrauenswürdig und können zur Bewertung der Attesting Party herangezogen werden. Der Datentransfer sollte natürlich über eine sichere Verbindung, z.B. SSL, erfolgen.
3) Der Integrity Validation Mechanism: Nachdem nun die Informationen vertrauenswürdig und unverfälscht beim Verifier angekommen sind, müssen sie geprüft werden, um den eigentlichen Zustand der Attesting Party festzustellen. Hierzu existiert auf dem Verifier System eine Datenbank mit verschiedenen Fingerprints, die mit den erhaltenen Measurement Listeneinträgen verglichen werden. Es existiert eine Policy, die den Umgang mit unbekannten oder nicht vertrauenswürdigen Fingerprints regelt. Wird eine Übereinstimmung mit einem nicht vertrauenswürdigen Eintrag festgestellt, wird die Attesting Party meist als nicht vertrauenswürdig eingestuft. Die Datenbank kann und muss immer wieder aktualisiert werden, um beispielsweise neue Software mit aufzunehmen oder die Fingerprints gepachter Versionen zu aktualisieren. Es besteht auch die Möglichkeit, Fingerprints, die von vertrauenswürdigen Dritten als nicht schädlich eingestuft und entsprechend zertifiziert wurden, mit in die Datenbank zu integrieren.

Remote Attestation ist nur eine Möglichkeit, die Funktionen einer Trusted Plattform zu nutzen. Es gibt noch viele weiter Anwendungen für die Trusted Computing eingesetzt werden könnte. Allerdings kommt unweigerlich die Frage auf, ob durch dieses Konzept nicht ein zu hohes Maß an Kontrolle ermöglicht wird, welches Software- oder Dienstanbieter missbrauchen könnten, um ihre Kunden zu überwachen. Einen kleinen Einblick in diese Frage und in die Argumente der Gegner und der Befürworter, soll das nächste Kapitel geben.

VII. Pro und Kontra Trusted Computing

„A trusted computer is a computer that can break my security" [16], so endet Ross Andersons FAQ über Trusted Computing. Mit seinem Paper, in dem er als erster Kritik an Trusted Computing übt, hat er eine große Widerstandswelle ausgelöst. Er weckte die Befürchtung, dass Trusted Computing dazu entwickelt wurde, um die Kontrolle über Trusted Plattformen zu gewinnen. Seiner Meinung nach sei die Hauptmotivation bei der Entwicklung der TCG-Spezifikationen das Digital Rights Management gewesen und das Ziel, Softwarepiraterie zu bekämpfen. Über eine Trusted Plattform ist es möglich, festzustellen, ob eine gültige Lizenz für eine Software oder für Dateien (z.B. Musik oder Filme) vorliegt. Ist dies nicht der Fall, kann die Ausführung dieser Inhalte unterbunden oder diese sogar gelöscht werden. Das sog. *Traitor Tracing* soll Raubkopien über Wasserzeichen erkennen und entfernen und das dafür verantwortliche System auf eine Blacklist setzen. Ross betont auch die Schwierigkeiten, die bei Trusted Computing Systemen entstehen können, wenn man alternative Software nutzen möchte. Um umzusteigen und alte Dateien mit der neuen Software bearbeiten und nutzen zu können ist immer eine Zustimmung der ursprünglichen Dateibesitzer notwendig, was den Wechselaufwand stark erhöht. So, laut Ross, kann z.B. Microsoft seine Marktstellung noch mehr stärken und Preise dirigieren. Seine größte Sorge sind die Zensurmöglichkeiten, die Trusted Computing mit sich bringt. Mit den oben erwähnten Blacklists könne nicht nur Piratensoftware gebannt, sondern auch politisches Material kontrolliert und verboten werden, was eine erhebliche Einschränkung der Freiheit wäre. Andere, wie Arbaugh [1], versuchen, Trusted Computing im Gesamtzusammenhang zu betrachten und auch die positiven Aspekte und Möglichkeiten für die Computersicherheit zu berücksichtigen. Arbaugh beispielsweise sieht das Problem eher im Privacy Bereich, da eine Trusted Plattform eindeutig ist und sie somit immer genau dem Besitzer zugeordnet werden kann. Auch dass Konzept der Attestation Identity Keys ist (wie bereits in Abbschnitt VI-A erwähnt) in in seinen Augen noch keine Lösung, da trotz allem die Zertifizierungsstelle, die für die Zertifizierung der AIK's zuständig ist, die einzelnen Schlüssel immer noch der Plattform und somit dem Besitzer zuordnen kann, da sie Informationen wie das EK Credential erhält.

Auf die viele Kritik antworteten die Entwickler, die an Trusted Computing beteiligt waren mit einem Rebuttal [17], in dem sie Ross's Argumente und die anderer Kritiker widerlegten und ihre Technologie verteidigten. Sie argumentierten, dass Begriffe wie Trusted Computing und DRM synonym verwendet worden sind, jedoch eine klare Trennung zwischen diesen Technologien besteht und diese auch sonst keine direkte Verbindung haben. Außerdem wurden Spekulationen über Trusted Computing gemacht, welche als Tatsachen dargestellt wurden, in Wirklichkeit aber so gar nicht in den Spezifikationen vorhanden sind. Sie kritisierten auch, dass die Papers voll von Fehlern in Bezug auf das technische Verständnis der Spezifikationen seien und dadurch viele Missverständnisse und falsche Annahmen aufkamen. Trusted Computing hat als primäres Ziel, dem Benutzer sicheres Schlüssel- und Datenmanagement zur Verfügung zu stellen und will in keinster Weise Einfluss auf seine Rechte zu nehmen. Das, was Systeme wie DRM oder das von Microsoft entwickelte NGSCB (Next-Generation Secure Computing Base) daraus machen, hat nichts mit dem Konzept des Trusted Computing an sich zu tun.

So entstand ein Hin- und Her zwischen Gegnern und Befürwortern. Die Quellen [1], [16], [17] und [18] bieten einen guten Anfang, um tiefer in die Diskussion einzusteigen.

VIII. Ausblick und Fazit

Nach einer Statistik von IDC[7] sind bis heute schon über 50 Millionen Systeme mit der notwendigen Technologie ausgestattet um als Trusted Plattform agieren zu können. Jedoch sind die TPMs standardmäßig abgeschalet und müssen vom Benutzer der Plattform erst explizit aktiviert werden um verwendet werden zu können. Ob dies jedoch auch getan wird ist eine andere Sache da viele Nutzer noch nicht überzeugt davon sind (s. Abschnitt VII) und auch erst wenige Anwendungen (neben der Remote Attestation) für Trusted Plattformen existieren. Einen großen Schritt zur Nutzung hat Microsoft mit dem *BitLocker* Konzept getan. [7] Dieses ist in das Windows Vista Betriebssystem integriert und bietet sichere Festplattenverschlüsselung mit Hife eines TPM an.

Trotz der hardwareunterstützten Sicherheitsmechanismen von Trusted Computing gibt es immer noch Möglichkeiten, diese zu umgehen. In [11] wird erfolgreich demonstriert, dass durchaus auch die Hardwarekomponenten eines TPM Schwachstellen haben können und somit angreifbar sind. Um sich im (v.a. Privat-)Anwenderbereich etablieren zu können, müssen Entwicklungen noch viel gegen die möglichen Verletzungen der Privatsphäre und die damit verbundenen Ängste, die mit Trusted Computing zusammenhängen, tun. Ein Ansatz ist z.B. die Virtualisierung, die heutzutage eine sehr große Rolle spielt. Ein rein virtuelles TPM wird in [19] vorgestellt. Virtuelle Maschinen besitzen dadurch die Möglichkeit, Trusted Computing in Anspruch zu nehmen. Vielleicht kann auf diese Weise ein Nutzer wieder die komplette Kontrolle über seine Plattform erlangen und trotzdem die Vorteile einer Trusted Plattform nutzen.

Vor allem aufgrund der hitzigen Diskussion um Trusted Computing ist ersichtlich, dass noch viele Fragen und Probleme

[7] http://www.itseccity.de/?url=/content/dailynews/090305_dailynews_text.html zuletzt besucht am 05.03.2009

offen sind. Wie bei so vielen Innovationen kommt es auch beim Trusted Computing darauf an, dass es richtig verwendet wird, dann kann es durchaus große Fortschritte im Bereich Sicherheit mit sich bringen. Der Einsatz in Unternehmensinfrastrukturen ist sicherlich sinnvoll, weil so eine sehr gute und zuverlässige Prüfung der einzelnen Rechner im und außerhalb des Netzes ermöglicht wird; dass diese immer aktuell sind und keine Schadsoftware darauf vorhanden ist. Sobald aber dadurch der Benutzer überwacht wird und somit eine Einschränkung seiner Freiheit erfährt, werden die positiven Aspekte schnell zu Negativen. Deshalb muss auf die richtige Verwendung der verfügbaren Technologien geachtet werden.

LITERATUR

[1] Willian A. Arbaugh. The tcpa; what's wrong; what's right and what to do about it. Technical report, University of Maryland, 2002.

[2] Sundeep Bajikar. Trusted platform module (tpm) based security on notebook pcs - white paper. Technical report, Mobile Platforms Group, Intel Corporation, 2002.

[3] Nicholas Bohm Brian Gladman, Carl Ellison. Digital signatures, certificates and electronic commerce, 1999.

[4] Claudia Eckert. *IT-Sicherheit, Konzepte-Verfahren-Protokolle*. Oldenbourg, 5. edition, 2007.

[5] Trusted Computing Group. Backgrounder, more secure computing, 2006. http://www.trustedcomputinggroup.org.

[6] http://de.wikipedia.org. Trustd computing platform alliance. zuletzt besucht am 22.04.2009.

[7] Jan Trukenmüller Jan-Peter Stotz Sven Türpe Jan Steffan, Andreas Poller. *BitLocker Drive Encryption im mobilen und stationären Unternehmenseinsatz*. Fraunhofer-Institut für Sichere Informationstechnologie.

[8] William J. Caelli Jason F. Reid. Drm, trusted computing and operating system architecture. Technical report, Information Security Research Center, Queensland University of Technology, 2005.

[9] Ed Dawson Eiji Okamoto Jason Reid, Juan M. Gonzalez Nieto. Privacy and trusted computing. Technical report, Information Security Research Center, Queensland University of Technology.

[10] Steve Johnson. Trusted boot loader. Technical report, Chair Security WG, Panasonic, 2006.

[11] Bernhard Kauer. Oslo: Improving the security of trusted computing. Technical report, Technische Universität Dresden.

[12] Thomas Müller. *Trusted Computing Systeme*. Springer Verlag Berlin Heidelberg, 2008.

[13] Siani Pearson. Trusted computing platforms, the next security solution. Technical report, HP Laboratories Bristol, 2002.

[14] James P. Ward Reiner Sailer, Leendert Van Doorn. The role of tpm in enterprise security. Technical report, Thomas J. Watson Research Center, 2004.

[15] Trent Jaeger Leendert van Doorn Reiner Sailer, Xiaolan Zhang. Design and implementation of a tcg-based integrity measurement architecture. Technical report, IBM T.J. Watson Research Center, 2004.

[16] Anderson Ross. 'trusted computing' frequently asked questions, 2003. http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html zuletzt besucht am 05.03.2009.

[17] David Safford. Clarifying missinformation on tcpa. Technical report, IBM Research, 2002.

[18] Seth Schoen. Trusted computing: Promise and risk.

[19] Kenneth A. Goldman Ronald Perez Reiner Sailer Leendert van Doorn Stefan Berger, Ramón Cáceres. vtpm: Virtualizing the trusted platform module. Technical report, IBM T.J. Watson Research Center, 2006.

[20] Trusted Computing Group. *TCG Software Stack (TSS)*, 2007. Version 1.2.